Properties and State

Overview

In this lab you'll enhance the "library" web page from the previous lab in 2 ways:

- You will use the React "propTypes" mechanism to specify tighter information for properties for each component (i.e., what data types are the properties, which properties are mandatory, and what default values might be appropriate).
- You'll add stateful behaviour to the web application, so that users can "like" the web page. You will hold the number of likes in mutable state.

You can either choose to implement your components as class-based components or as functional components. We provide starter code and solutions for both techniques.

Source folders

- ReactDev\Student\05-PropertiesAndState
- ReactDev\Solutions\05-PropertiesAndState

Roadmap

Here's a brief summary of the exercises in this lab. More detailed instructions follow later in this lab document:

1. Familiarization with the 'solution' web pages
2. Getting started with the 'student' web page
3. Specifying property type information
4. Adding stateful behaviour to the web application
5. (If time permits) Defining a Library component

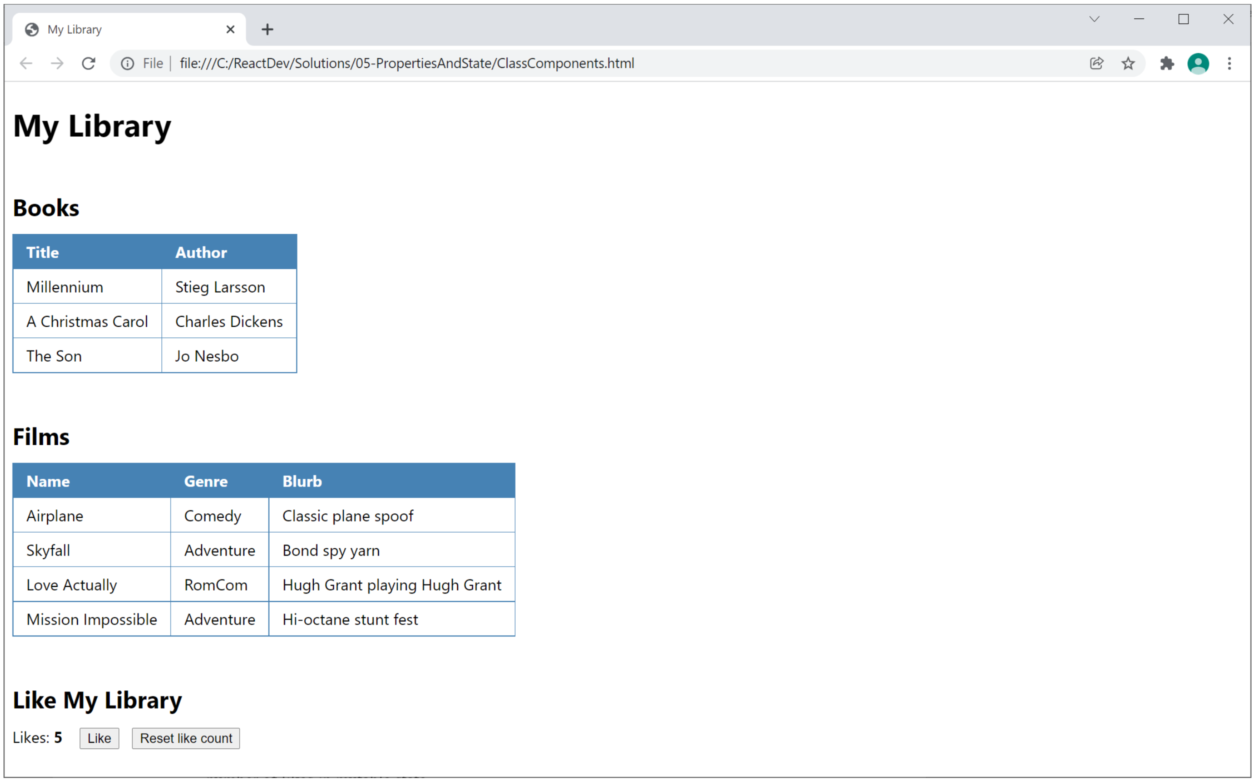
Exercise 1: Familiarization with the 'solution' web pages

Open either of the following web pages in a browser:

ReactDev\Solutions\05-PropertiesAndState\ClassComponents.html

ReactDev\Solutions\05-PropertiesAndState\FunctionalComponents.html

The UI is slightly different to before:


My Library
File
file:///C:/ReactDev/Solutions/05-PropertiesAndState/ClassComponents.html

My Library

Books

Title	Author
Millennium	Stieg Larsson
A Christmas Carol	Charles Dickens
The Son	Jo Nesbo

Films

Name	Genre	Blurb
Airplane	Comedy	Classic plane spoof
Skyfall	Adventure	Bond spy yarn
Love Actually	RomCom	Hugh Grant playing Hugh Grant
Mission Impossible	Adventure	Hi-octane stunt fest

Like My Library

Likes: **5** Like Reset like count

Note the following points:

- The web page only displays tabular info in this screenshot. In fact, we've wrapped the UI in a <Library> component, and this component receives a tabularFormat property. If this property is true, the <Library> component renders books and films as HTML tables, otherwise it renders the books and films as HTML lists.
- At the bottom of the web page, there's a panel that allows the user to "like" the web site. Click the *Like* button to increment the number of likes, or click the *Reset like count* button to reset the number of likes to 0. The web application remembers the current number of likes in mutable state.

Exercise 2: Getting started with the 'student' web page

Now go to the ***student*** folder and open either of the following web pages in a text editor:

ReactDev\Student\05-PropertiesAndState\ClassComponents.html

ReactDev\Student\05-PropertiesAndState\FunctionalComponents.html

These files are the same as the solutions from the previous lab. Take a moment to get familiar with the code.

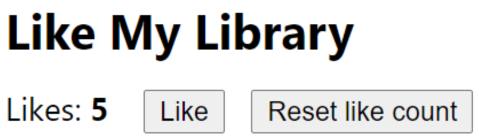
Exercise 3: Specifying property type information

Add property type information for all the properties for each component in your web application. For each property, specify:

- Its data type
- Whether it's required or optional
- A default value, if appropriate

Exercise 4: Adding stateful behaviour to the web application

Define a new component named LikePanel, to allow the user to "like" your web site:



Like My Library

Likes: **5**

Hints and requirements:

- The LikePanel component needs to maintain the likes count in mutable state.
- When the user clicks the *Like* button, increment the likes count in mutable state.
- When the user clicks *Reset like count*, reset the likes count to 0 in mutable state.

Remember to render LikePanel as part of the overall web application. When you've done all this, open the web page in a browser and verify it all works.

Exercise 5 (If time permits): Defining a Library component

Refactor the web page so that the overall UI is rendered by a Library component, as follows:

- The Library component needs 3 properties (remember to specify full "propTypes" info for all these properties):
  - books - an array of books to display (required)
  - films - an array of films to display (required)
  - tabularFormat - an optional bool indicating whether to display tabular data
- The component renders the books and films as HTML tables (if the tabularFormat property is true) or as HTML lists otherwise.